International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 5, Oct 2014, 91-100
© TJPRC Pvt. Ltd.

THE EFFECT OF SIMILARITY MEASURES ON GENETIC ALGORITHM-BASED INFORMATION RETRIEVAL

MOHEB RAMZY GIRGIS, ABDELMGEID AMIN ALY & FATIMA MOHY ELDIN AZZAM

Department of Computer Science, Faculty of Science, Minia University, ElMinia, Egypt

ABSTRACT

Genetic algorithms (GAs) can be used in information retrieval (IR) to optimize the query "solution". This paper proposes a GA-based IR algorithm that adjusts the weights of keywords of a query in order to generate an optimal or near optimal query vector. In this algorithm, each query is represented by a chromosome. These chromosomes are feed into genetic operator process: selection, crossover, and mutation to get new population, then, to get better solutions, a local search procedure is applied on each individual in the new population. This process is repeated until an optimized query chromosome for document retrieval is obtained. The evolution of the possible solutions is guided by fitness functions that are designed to measure the goodness of those solutions. We used order-based fitness function with different similarity measures to study their effect on the quality of the generated solutions and decide which similarity measure leads to the best solution.

KEYWORDS: Information, Retrieval, Genetic Algorithm, Query Optimization, Similarity Measures

I. INTRODUCTION

Information retrieval deals with the representation, storage, organization, and access to information items [1]. Information retrieval aims at retrieve all objects which satisfy clearly defined conditions such as those in a regular expression or in a relational algebra expression. The representation and association of the information items will provide the user with effortless access to the information in which he/she will be interested.

Genetic Algorithms (GAs) are probabilistic search methods that have been developed by John Holland in 1975 [2], [3]. GAs applied natural selection and natural genetics in artificial intelligence to find the globally optimal solution to the optimization problem from the feasible solutions. It is often used as an optimization method to solve problems where little is known about the objective function. The operation of the GA is quite simple. It starts with a population of random individuals, each corresponding to a particular candidate solution to the problem to be solved. Then, the best individuals survive, mate, and create offspring, originating a new population of individuals. This process is repeated a number of times, and typically leads to better and better individuals. Nowadays GAs have been applied to various domains, including timetable, scheduling, robot control, signature verification, image processing, packing, routing, pipeline control systems, machine learning, and information retrieval.

This paper proposes a GA-based IR algorithm that adjusts the weights of keywords of a query in order to generate an optimal or near optimal query vector. In this algorithm, each query is represented by a chromosome. The algorithm performs the genetic operator process: selection, crossover, and mutation on the current population to get new population, then, to get better solutions, a local search procedure is applied on each individual in the new population. This process is repeated until an optimized query chromosome for document retrieval is obtained. The evolution of the possible solutions is guided by fitness functions that are designed to measure the goodness of those solutions. We have used order-based fitness function with 5 different similarity measures to study their effect on the quality of the generated solutions and decide which similarity measure leads to the best solution.

The paper is organized as follows: Section 2 gives a review of related work. Section 3 describes the steps of the main process a GA. Section 4 presents the problem statement. Section 5 described the components of the proposed GA for information retrieval: chromosome representation, genetic operators, fitness function, and local search procedure. Section 6 presents the results of the experiments that have been conducted to test queries with different similarity measure functions and different mutation probability values. Section 7 presents the conclusion of this research work.

II. RELATED WORK

There are several studies that used GA to optimize the user query in information retrieval system (IRS). Radwan et al. [4] investigated the use of GAs in information retrieval. They presented a new fitness function to approximate information retrieval which is very fast and very flexible than the cosine similarity fitness function. Chaudhary and Suri [5] discussed the impact of optimization using GA and share GA on multimodal image registration by considering mutual information concept. Bhatnagar and Pareek [6] discussed the applications of GA for improving retrieval efficiency of IRS. GA was used to find an optimal set of weights for components of combined similarity measure consisting of different standard similarity measures that are used for ranking the documents. Aly [7] presented an adaptive method using GA to modify user’s queries, based on relevance judgments. Vrajitoru [8] introduced a new crossover operation during the implementation of the GA in IR in order to assist IRS to find, in a huge text documents collection, a good reply to a query expressed by the user, and he compared it with other learning methods. Al Mashagba et al. [9] used GA to find the best strategy and fitness function that can be used when the data collection is in the Arabic language, so they used different similarity measures in vector space model and for each similarity measure different GA using different crossover and mutation. Dashti and Zad [10] presented a method using GA in a distributed way according to users' favorites to optimize query sent to search engine and finally to optimize quality of result pages. Mercy and Naomie [11] proposed a framework of data fusion approach based on linear combinations of retrieval status values obtained from Vector Space Model and Probability Model system. They used GA-based approach to find the best linear combination of weights assigned to the scores of different retrieval systems to get the most optimal retrieval performance. Nassar et al. [12] described optimization technique to optimize user query in Arabic data. To optimize query, they used GA with different fitness, different crossover and mutation technique. This technique are applied in Boolean model. Sathya and Simon [13] described document crawler which is used to extract information from web database. As volume of data in web is large they used GA to extract relevant data. Three main steps to extract data from database were proposed. First extract data using document crawler, second applying GA to get relevant data, and third applying result from GA to IRS to get better result. Ibrahim et al. [14] presented a model of hybrid GA-Particle Swarm Optimization (HGAPSO) based query optimization for Web information retrieval. The keywords are used to produce new keywords that are related to the user search. Sihombing et al. [15] compared Horng and Yeh formulation [16] in IRS with Jaccard and Dice similarity measures, All the 3 techniques are implemented in IRS using GA. Zhu et al. [17] described relevance feedback technique to retrieve relevant information.

A GA to optimize user query and retrieve web information was applied. Vicente P. and Cristina P. [18] described various order based fitness functions than evaluate efficiency of GA using this fitness function for relevance feedback.

Simon and Sathya [19] described a general frame work of information retrieval system and they discussed the applicability of GA in different areas of information retrieval such as genetic mining, query optimization, document clustering, and query optimization etc. Pathak et al. [20] described a method that applied GAs to adapt various matching functions. Such adaptation of the matching functions led to a better retrieval performance than that obtained by using a single matching function.

III. PROCESS OF GENETIC ALGORITHMS

Figure 1 illustrates the main process a GA, which aims at finding the best solution for the given problem. As shown in the figure the GA process includes the following steps:

- **Initialization:** A population of chromosomes, representing initial solutions of the given problem, is created randomly at the beginning.
- **Selection**: The fitness value of each chromosome in the current population is evaluated. The chromosomes with better fitness values are selected into the recombination pool using a selection method, such as the roulette wheel or the tournament selection method.
- **Crossover:** Some genes of two parent chromosomes are exchanged to obtain new offsprings in an attempt to get better solutions.
- **Mutation:** Mutation operation is applied to change some genes.
- **Evolutionary Cycle**: A termination criterion is used to determine if the process should terminate or the process should go to step 2 repeatedly for next generation.

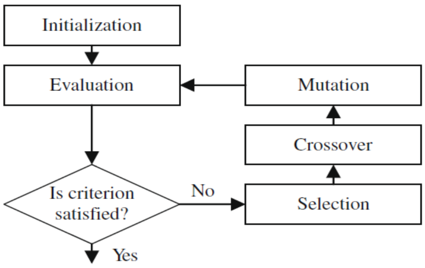


Figure 1: The Process of GA

IV. PROBLEM STATEMENT

Recently, people have started dealing with an increasing number of electronic documents in information networks. Finding the documents that users need from all the available documents is an important issue. An approach to solve this problem is to adapt the query vector in order to retrieve as many relevant documents as possible, and this is achieved by many ways. Indeed, the main differentiating feature of each of those ways is the fitness function used. The best results were obtained with the functions that took into account not only whether the possible solution retrieves many relevant and few irrelevant documents, but also whether the relevant documents were given at the beginning of the list or at the end.

This study used GA with non-interpolated average precision as a fitness function in order to optimize the query. We used different similarity measures and mutation probabilities to obtain the best technique that will generate the most optimized query.

V. THE PROPOSED GA-BASED IR APPROACH

The components of the proposed GA for information retrieval can be described in the following subsections.

1. Chromosome Representation

The model used in this paper is represented by a double-tuple (T, W), where T= (t_1, t_2, …, t_i, ..., t_n), t_i represents the i^{th} keyword (feature term), and n is the number of keywords; and W = (w_1, w_2, . . ., w_i, . . ., w_n), where w_i represents the weight of the i^{th} keyword t_i. The weight vector is also called a query vector [21]. In the proposed GA, a chromosome with a length *n* represents a query vector, where the genes are real numbers representing the keywords weights.

2. Document Representation

A document is represented as a vector D (x_1, x_2,..., x_n), where x_i denotes the frequency of the term (keyword) t_i in the document D. The frequency x_i is defined as follows: $x_i = n_i/n_{all}$, where n_i represents the occurrence of the term t_i in the document D, and n_{all} is the total number of terms in the document D.

3. The Fitness Functions

For each problem to be solved, one has to supply a fitness function, *f*, which evaluate how good each solution be. The information retrieval problem is how to retrieve user required documents. In the considered IR problem, the fitness function is based on the similarity between the documents and the query.

There are 2 types of similarity measure functions as shown in Table 1: weighted term vector and binary term vector, where X = (x_1, x_2, x_3,….., x_n), | X | = number of terms occur in X, | X ∩Y | = number of terms occur in both X and Y, X represents the document vector, and Y represents the query vector.

The fitness function used here is the non-interpolated average precision [22], [23] that is constructed as follows:

Firstly, calculate the similarity of the query vector with all the documents (using a similarity measure function). Secondly, sort the documents into decreasing order according to similarity. Finally, calculate the fitness value of the chromosome using the following formula:

$$\mathrm{AvgP} = \frac{1}{|D|} \sum_{i=1}^{|D|} r(d_i) \sum_{j=i}^{|D|} \frac{1}{j} \qquad (1)$$

where function r(d) gives the relevance of a document d. It returns 1 if d is relevant, and 0 otherwise, |D| represents the number of training documents.

In the proposed GA approach we used the different similarity measure functions mentioned in Table 1.

Table 1: Similarity Measure Functions

Similarity Measure Sim (X,Y)	Binary Term Vectors	Weighted Term Vectors
Inner product	$\lvert X \cap Y \rvert$	$\sum_{i=1}^{n} x_i \cdot y_i$
Dice coefficient	$2\,\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$	$2\,\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2}$
Cosine Coefficient	$\frac{\lvert X \cap Y \rvert}{\sqrt{\lvert X \rvert . \lvert Y \rvert}}$	$\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 . \sum_{i=1}^{n} y_i^2}}$
Jaccard Coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert - \lvert X \cap Y \rvert}$	$\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i \cdot y_i}$
Overlap Coefficient	$\frac{\lvert X \cap Y \rvert}{\min(\lvert X \rvert, \lvert Y \rvert)}$	$\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\min(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i)}$

4. Selection Process

In the selection mechanism, the GA uses "simple random sampling" [24],[3]. This consists in constructing a roulette with the same number of slots as there are individuals in the population, and in which the size of each slot is directly related to the individual's fitness value. Hence, the best chromosomes will have more chance to survive and contribute to the next generation than the worst chromosomes.

5. Genetic Operators

Crossover*:* The crossover operator selects two parents to crossover and generates one offspring. The parent chromosomes are chosen by selecting a uniform random number between 0 and 1, if the random number is less than the crossover probability (P_c), the chromosome is selected to the crossover operation that is described below:

Let Q_x and Q_y be two selected parent chromosomes. The offspring Q_z is constructed under the assumption AvgP(Q_x)>AvgP(Q_y), The offspring Q_z is generated as follows:

$$Q^z = (w_1^z, w_2^z, w_3^z, \ldots\ldots\ldots, w_n^z)$$

$$w_i^z = w_i^x + (\, w_i^x - w_i^y)$$

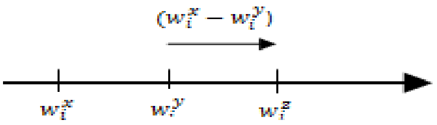


Figure 2: The Behavior of the Natural Crossover Operator

Mutation: In the proposed algorithm, mutation chooses a random gene according to the mutation probability (p_m) and replaces its value with a different one according to the following formula:

$$w_i^{old} - w_i^{best} = w_i^{best} - w_i^{new}$$

where w_i^{old} is the weight of the selected gene in the chromosome; w_i^{best} is the weight of the best gene in the population; w_i^{new} is the new weight of the gene in the selected chromosome.

6. Local Search

Local Search Procedures (LSPs) are optimization methods that maintain a solution, known as current solution, and explore the search space by steps within its neighbourhood. They usually go from the current solution to a better close solution, which is used, in the next iteration, as current solution. This process is repeated till a stop condition is fulfilled, e.g. there is no better solution within the neighbourhood of the current solution.

To generate a neighbour of a query vector Q_j, the value of a gene is to be increased or decreased. This approach is usually used in GAs [24]. Let and be the neighbors of the query vector Q_j, where j ranges from 1 to n:

$$Q_j = (w_1, w_2, w_3 \dots\dots\dots w_n)$$

$$Q_j^+ = (w_1^+, w_2^+, w_3^+ \dots\dots\dots w_n^+)$$

$$Q_j^- = (w_1^-, w_2^-, w_3^- \dots\dots\dots w_n^-)$$

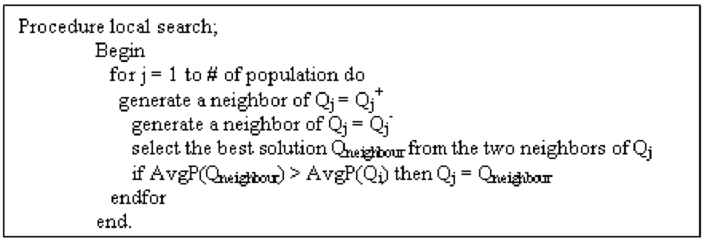
The weights in Q_j^+ and Q_j^- are defined below:

$w_k^+ = w_k$ if $1 \le k \le n$ and $k \le j$

$w_k^+ = w_k * (1 + \alpha)$ if $1 \le k \le n$ and $k = j$

$w_k^- = w_k$ if $1 \le k \le n$ and $k \le j$

$w_k^- = w_k * (1 - \alpha)$ if $1 \le k \le n$ and $k = j$



```
Procedure local search;
        Begin
          for j = 1 to # of population do
            generate a neighbor of $Q_j = Q_j^+$
              generate a neighbor of $Q_j = Q_j^-$
              select the best solution $Q_{neighbour}$ from the two neighbors of $Q_j$
              if AvgP($Q_{neighbour}$) > AvgP($Q_j$) then $Q_j = Q_{neighbour}$
          endfor
        end.
```

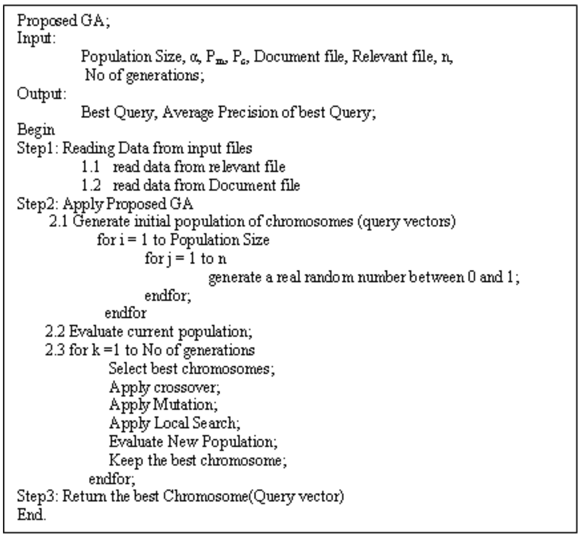
Figure 3: Local Search Procedure

where α ranges from 0 to 1. The value of α decides the ratio of increase or decrease. Each query vector generates two neighbouring vectors. From all neighbouring vectors, the vector $Q_{neighbour}$ which has the best fitness function value is selected. If AvgP($Q_{neighbour}$) is larger than AvgP(Q_j), the vetor Q_j is replaced. The local search procedure is shown in Figure 3.

7. The Overall Algorithm

The proposed overall GA-based IR algorithm is described in Figure 4. This algorithm adjusts the weights of keywords in order to generate an optimal or near optimal query vector. To achieve this aim, the algorithm applies the local search procedure on each individual in the new population in each generation. The algorithm uses two input files:

the document file and the relevant file, where the document file contains the documents information, i.e. terms that each document contains and the frequency of each term. The relevant file contains data that determines the relevant and irrelevant documents.



```
Proposed GA;
Input:
          Population Size, $\alpha$, $P_m$, $P_c$, Document file, Relevant file, n,
           No of generations;
Output:
          Best Query, Average Precision of best Query;
Begin
Step1: Reading Data from input files
          1.1   read data from relevant file
          1.2   read data from Document file
Step2: Apply Proposed GA
     2.1 Generate initial population of chromosomes (query vectors)
            for i = 1 to Population Size
                   for j = 1 to n
                             generate a real random number between 0 and 1;
                   endfor;
             endfor
     2.2 Evaluate current population;
     2.3 for k =1 to No of generations
              Select best chromosomes;
              Apply crossover;
              Apply Mutation;
              Apply Local Search;
              Evaluate New Population;
              Keep the best chromosome;
           endfor;
Step3: Return the best Chromosome(Query vector)
End.
```

Figure 4: Proposed GA-Based Information Retrieval

1. Experimental Results

Tables 2, 3 and 4 show the average precision of the optimized queries when the 5 different similarity measure functions were used with 3 different mutation probabilities: Pm = 0.01, 0.3 and 0.5, respectively.

The results of the experiments indicated that:

- The similarity measure function F4 "Jaccard coefficient" gave the best average precision in the three cases.
- Information retrieval with Pm = 0.01 yielded the highest average precision, followed by Pm = 0.3, then Pm = 0.5, which yielded the lowest average precision.

Table 2: Average Precision for 5 Similarity Measure Functions with PC = 0.8 and PM = 0.01

Field	F1	F2	F3	F4	F5
Object Oriented	.97	.97	.95	.97	.97
Operating System	.94	.93	.94	.96	.93
Computer Networks	.97	.97	.95	.98	.94
Average	.96	.95	.94	.97	.94

Table 3: Average Precision for 5 Similarity Measure Functions with PC = 0.8 and PM = 0.3

Field	F1	F2	F3	F4	F5
Object Oriented	.95	.95	.93	.96	.94
Operating System	.93	.92	.93	.96	.93
Computer Networks	.92	.85	.89	.92	.85
Average	.93	.90	.91	.94	.90

Table 4: Average Precision for 5 Similarity Measure Functions with PC = 0.8 and PM = 0.5

Field	F1	F2	F3	F4	F5
Object Oriented	.88	.88	.88	.88	.88
Operating System	.93	.91	.93	.95	.91
Computer Networks	.80	.79	.77	.82	.77
Average	.87	.86	.86	.88	.85

VI. CONCLUSIONS

This paper presented a proposed GA-based IR algorithm that adjusts the weights of keywords of a query in order to generate an optimal or near optimal query vector. In this algorithm, each query is represented by a chromosome. The algorithm performs the genetic operator process: selection, crossover, and mutation on the current population to get new population, then, to get better solutions, a local search procedure is applied on each individual in the new population. This process is repeated until an optimized query chromosome for document retrieval is obtained. The evolution of the possible solutions is guided by fitness functions that are designed to measure the goodness of those solutions.

We have used order-based fitness function with 5 different similarity measures to study their effect on the quality of the generated solutions and decide which similarity measure leads to the best solution.

Experiments were conducted with the developed IR system that implements the proposed algorithm to test queries with 5 different similarity measure functions: Inner product, Dice coefficient, Cosine coefficient, Jaccard coefficient, and Overlap coefficient, and with 3 different mutation probabilities. The results of the experiments indicated that the similarity measure function "Jaccard Coefficient" gave the best average information retrieval precision with the 3 different mutation probabilities, and the highest precision obtained when a small mutation probability was used.

REFERENCES

1. R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval,* 2nd ed, Harlow, England: Addison Wesley, 1999.
2. L. D. Davis, *Handbook of Genetic Algorithms*, 1st, New York: Van Nostrand Reinhold, 1991.
3. J. H. Holland, *Adaptation in natural and artificial systems,* 2nd ed, Cambridge, MA: MIT Press, 1992.
4. A. A. A. Radwan, B. A. Abdel Latef, A. A. Ali, O. A. Sadek, "Using Genetic Algorithm to Improve Information Retrieval Systems", in *Proc*. WASET, 2008, ISSN 1307-6884, p.756 .
5. V. Chaudhary, P. R. Suri, “Genetic algorithm v/s share genetic algorithm with roulette wheel selection method for registration of multimodal images”, *International Journal of Engineering Research and Application*, vol. 2, pp. 365-370, Aug. 2012.
6. P. Bhatnagar and N. K. Pareek, “A combined matching function based evolutionary approach for development of adaptive information retrieval system”, *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, pp. 249-256, Jun. 2012.
7. A. A. Aly, "Applying Genetic Algorithm in Query Improvement Problem", *International Journal Information Technologies and Knowledge*, vol.1, p 309-316. 2007.

8. D. Vrajitoru, “Crossover improvement for the genetic algorithm in information retrieval”, *Information Processing& Management*, vol. 34, pp. 405–415, Jul. 1998.

9. E. Al Mashagba, F. Al Mashagba and M. O. Nassar, “Query optimization using genetic algorithm in the vector space model”, *International Journal of Computer Science*, vol. 8, pp. 450-457, Sept. 2011.

10. F. Dashti, and S. A. Zad," Optimizing the data search results in web using Genetic Algorithm", *International journal of advanced engineering and technologies*, vol. 1, pp.16 -22, 2010.

11. M. T. bt. Mulyadi., N. Salim, "A Framework for Genetic-Based Fusion of Similarity Measures In Chemical Compound Retrieval", in *Proc*. International Symposium on Bio-Inspired Computing, Puteri Pan Pacific Hotel Johor, Sept. 2005.

12. M. O. Nassar, F. Al Mashagba and E. Al Mashagba, “Improving the user query for the boolean model using genetic algorithm”, *International Journal of Computer Science*, vol. 8, pp. 66-70, Sept. 2011.

13. S. S. Sathya and P. Simon, “A document retrieval system with combination terms using genetic algorithm”, *International Journal of Computer and Electrical Engineering*, vol. 2, pp.1-6, Feb. 2010.

14. N. A. Ibrahim, A. Selamat, M. H. Selamat, “Query optimization in relevance feedback using hybrid GA-PSO for effective web information retrieval”, in *Proc*. Third Asia International Conference on Modelling & Simulation, 2009, p. 91.

15. P. Sihombing, A. Embong, P. Sumari, “Comparison of document similarity in information retrieval system by different formulation”, in *Proc*. of 2nd IMT-GT Regional Conference on Mathematics Statics and Application, Jun. 2006.

16. j. -T. Horng and C. -C. Yeh, "Applying genetic algorithms to query optimisation in document retrieval", in *Proc*. Information Processing and Management, 2000, P. 737.

17. Z. Zhu, X. Chen, Q. Xie and Q. Zhu, “A GA based query optimization for web information retrieval”, in *Proc*. International Conference on Intelligent Computing, Aug. 2005, p. 2069.

18. V. P. –G. Bote, C. L. Pujalte and F. D. Anegon, "Order-Based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback", *Journal Of The American Society For Information Science And Technology*, vol. 54(2), pp.152–160, 2003.

19. P. Simon, and S. S. Sathya, “Genetic algorithm for information retrieval”, in *Proc*. International Conference on Intelligent Agent & Multi-Agent Systems (IAMA), 2009, p.1.

20. P. Pathak, M. Gordon and W. Fan. “Effective information retrieval using genetic algorithms based matching functions adaption”, in *Proc*. 33rd Hawaii International Conference on Science (HICS), 2000.

21. J. -J. Yang and R. R. Korfhage, "Query optimization in information retrieval using genetic algorithms". in *Proc*. Fifth International Conference on Genetic Algorithms, 1993, p. 603.

22. C. –H. Chang., and, C. -C. Hsu, "Information searching and exploring agent applying clustering and genetic algorithm", In *Proc*. 1st Agent Technology Workshop, 1997.

23. C. –H. Chang., and, C. -C. Hsu, "The design of an information system for hypertext retrieval and automatic discovery" on WWW. Ph.D. thesis, Department of CSIE, National Taiwan University.
24. D. E. Goldberg, *Genetic algorithms in search optimization and machine learning*, 1st ed, Reading, MA: Addison Wesley, 1989.